(Edited Jan. 14, 2022)

Introduction

In this CAD you will design a Register File (RF). This is the first module you will create that will be used in your final RISC microprocessor design.

Register File

For CAD3, you will design a 16-word, 16-bit RF with 1 write port and two read ports. One of the structures that is central to the datapath is the RF. An RF consists of a set of registers that can be read from or written to. Writing a register requires (i) an address, (ii) the data to be written & (iii) a signal that controls the write timing. Reading a RF can often be controlled by just the address but may include additional logic. The RF for this CAD is to include an array of flip-flops or latches and properly sized buffers for the associated read/write circuits. Address decode logic is not required in CAD3, but will have to be added to your design when the RF is integrated with the microprocessor controller.

Register Cell

Conventionally, a RF consists of an array of static RAM cells with read/write circuitry and a sense amplifier. Though this type of implementation yields a fast, compact design, it requires much design effort, especially for the sense amplifiers. In small RFs (few registers), the support circuitry for the SRAMs is used by a small number of registers, making the SRAM-based RF larger than some other types. An alternative approach is to use a latch consistent with your clocking scheme, modified to have two read ports and one write port. One such example is given in Figure 1. This method uses one master latch to supply data to 1 of the 16 slave latches for a particular write operation. The control signals that determine the timing of write/read operation are controlled by decoded signals from the control unit.

Drivers and Buffers

Control signals (which come from the control unit – CAD7) shared by all 16 bits of a register can have large capacitive loading compared to other signals in your design. So it's a good idea to have a strong driver for these signals. The data output busses on the RF would also have large loading associated with them. These signals are candidates for transistor sizing on the gates that drive them or possibly even the insertion of buffers. In either case, these design decisions should be based on simulations. When inserting drivers/buffers, be careful of the polarity of the signals/outputs.

Bus Strategy

You can use several methods to drive values onto the output busses. In this class you will use transmission-gate drivers on the outputs of each component that can

drive the bus. This will require only one bus, versus other strategies that would require multiple buses, one for each word line. Using transmission gate would also ensure that the integrity of the value stored on your bus remains even when the data is not being read. Using tri-state busses would require keeper latches so that the bus doesn't float to an intermediate value if the data is not being read. The schematic on the next page has transmission-gates, and you should recreate it for this CAD assignment.

Please see the documents "busnames.pdf" and "bustap.pdf" for a description of how to create busses within your schematics.

Metal Layers

Only use the following metal layers to do your routing: M1, M2, and M3. Metals 4 & 5 are being saved for global routing in the last CAD assignment. This means that you must plan your RF carefully so that you only route in the lower layer metals but are still able to keep the layout compact. Plan which metal tracks you will be using for routing.

Procedure

Schematic Design

Understand the read/write methodology by studying the schematic in Figure 1. You will not be responsible for designing the decoder for this CAD assignment. Remember to use w=160nm for "weak" transistors. Then, replicate your 16-bit register 16 times to form the array. You will need to design the driver circuits based on capacitance estimates (see "**Speed-Path Schematic**").

Speed-Path Schematic (Very Important!)

Create a special schematic for the purpose of sizing devices prior to starting layout. Since it can be difficult to resize devices after layout is complete, it makes sense (particularly in the custom design of array structures like memories or RFs) to simulate on a simple, representative schematic containing the basic master-slave latches, control signal buffers, and estimated parasitic capacitances. For example, let's say your design uses a full-CMOS transmission-gate for driving an output bus. One bit of that bus would therefore have 16 full-CMOS transmission gates (one for each register), parasitic routing capacitance, as well as any additional loading outside of the register file itself. The source/drain loading due to the transmission gates can be modeled easily with "dummy" transistors – transistors with gate and source tied to VDD (PMOS) or VSS (NMOS) and drain tied to the output node. Transistor gate loading can be modeled similarly by tying source/drain to VDD or VSS and connecting the gate of the "dummy" device to the node in question. Ballpark capacitance estimates for minimum width routes in this technology are: poly – 0.16fF/um, M1 – 0.25fF/um, M2 – 0.20fF/um, M3 – 0.20fF/um, M4 – 0.20fF/um, M5

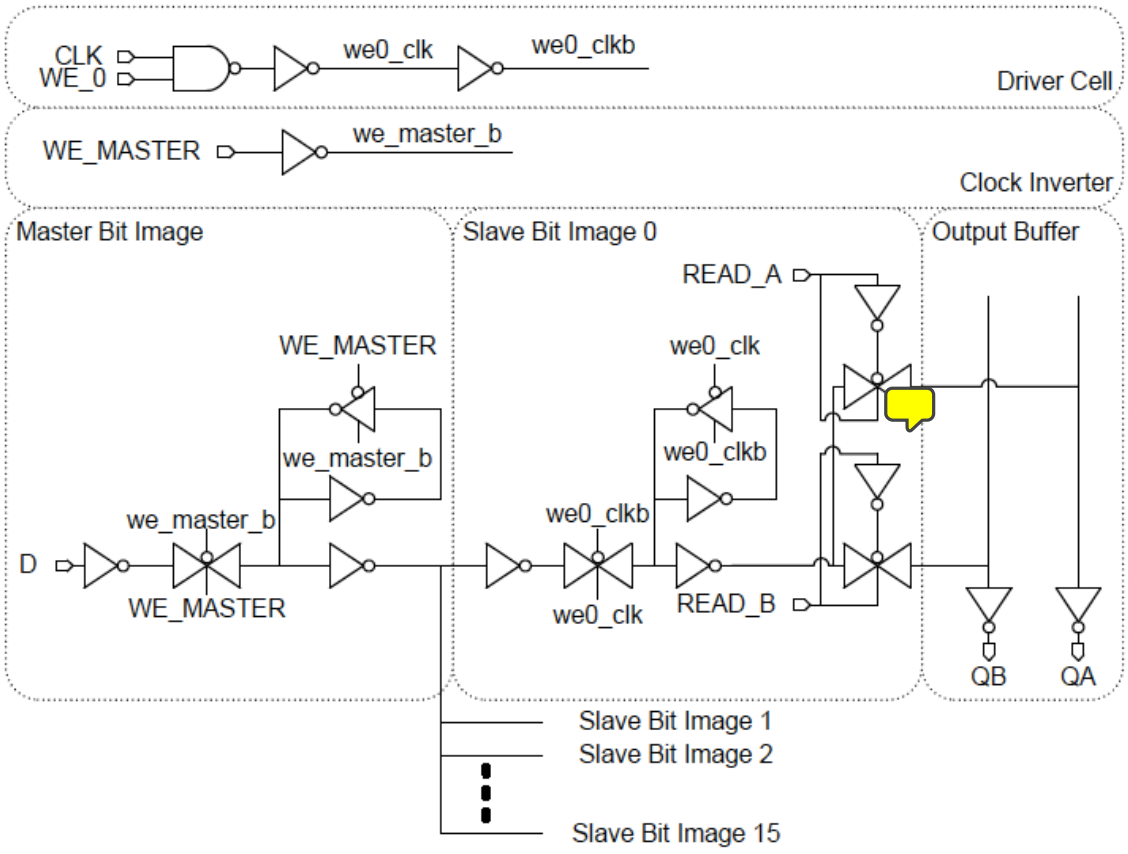

Figure 1: RF Schematic for ECS427 CAD3

- 0.20fF/um. The gate capacitance is 1.3fF/um for NMOS/PMOS with minimum gate length. Once you have an estimate for the routing capacitance, use the capacitor (CAP) symbol from "analogLib" library. You should use this modeling technique for **any nodes that look like they will be heavily loaded** in terms of routing length and the number of attached transistors. Capacitance of other nodes (e.g. internal latch nodes) will be dominated by transistor loading, which is already accounted for by the transistor model. With this schematic in place, you can run simulations to intelligently size devices in your control signal buffers and devices along your output path. As for target delays, the requirements will vary depending on your group's application. In one clock cycle, your controller will have to decode the instruction word, assert read signals to the RF, the RF will provide operands to the ALU, the ALU will perform an operation and have data ready for writing back to the RF by the end of the cycle. This is the leading candidate for the critical path in these designs. Sometimes memory accesses are the critical path. 100MHz (10ns period) is a relatively easy target frequency to meet with this technology. 400MHz (2.5ns period) is about the absolute maximum for this technology. You do not need to try to meet this requirement in your EECS 427 microprocessor, however it is a useful benchmark to keep in mind.

NCVerilog

Run NCVerilog on the RF and verify that it functions as expected by running a few test sequences. For example, write to each location and then read from each location. You must note that if you perform a write operation and then change the write address, you will write the same master-latch value at the new address. You can deal with this when you design the controller in CAD7.

Layout

Since the RF is one of the key modules of your datapath, care is needed to minimize the parasitic capacitances on critical signal nets, as this directly affects the performance of your processor. Keeping the structure dense and compact while avoiding long interconnects is a prime goal. Mirroring a cell and thereby sharing wells and the supply lines is one of the most frequently used strategies. You should design the floorplan and routing strategy before you start on your layout. Power lines in the cells should be dimensional so that they can carry the current required by the row of connected cells. As good ballpark figure for this process is to keep average current densities below 1mA/um of metal width. (Power rings/grids not required for this CAD.) Use hierarchy effectively to build up your register file layout. The use of hierarchy would also help you isolate DRC and LVS failure since DRC and LVS check are done in small increments.

Design Verification

You may end up spending a lot of time on verification if you have errors. The best way to avoid this is to check the design thoroughly as you go along. It is much better to spend 5 minutes in checking the circuit, than to waste 30 minutes debugging the RF macro later. Run DRC, LVS, on the entire RF. Extract the parasitics and back-annotate those into your speedpath to get a more accurate read and write time.

Delay Estimation

Now that your layout is complete and PEX has been run, you can add any additional capacitance for loading outside of the RF and derive more realistic delay estimates. There are two ways to do this. You could reuse your speedpath schematic or simply simulate the entire RF. If you reuse your speedpath schematic, you will have to get “real” capacitance information from your back-annotation and place or update capacitors in your speedpath. If you run SPICE on the entire RF, force inputs such that only location is being read. Note that the worst-case in the tri-state/transmission gate bus architecture is when one register is read out onto both output busses simultaneously (e.g. ADD r0, r0 would trigger this path).

Comments & Requirements

- Since you are working in groups now, you only need to submit one copy of the required register file for grading. We will identify groups by group

number. Do your CAD3 work in that group directory. For example, if you are group1, your class directory will be: /afs/umich.edu/class/eecs427/w22/group1.

- Try to think about the routing strategy in advance – how, and where to use M1, M2, and M3. It is ok to occasionally use M2 and M3 in your leaf cell (lowest-level) layout but limit their use as much as possible and stick to the required routing directions.
- Plan your team's time in advance. Meet early, decide on the basic circuit to implement and decide who will do what and when. **Don't wait until a few days before the deadline to start**. Get started right away.
- While you are working on CAD3, think about several other modules in your datapath. Try to understand where your RF fits into the datapath, especially if you do not have any experience with basic computer architecture & pipelines.
- How are you going to integrate with your RF? How are your busses going to run through your datapath? The more resources you have for routing data signals over the RF, the more flexibility you will have in CAD7 when placing and routing the datapath components together. Making your data outputs and data inputs easily accessible from top AND bottom of the RF will be helpful.
- Shoot for a compact layout. Success in this area will depend almost entirely on the layout of the array (slave) cell, which will be replicated 256 times. Remember bit-slice width target of 4.4-6.4um. As always, your outline height and width should be multiples of 0.4um.
- Only use M1, M2, and M3 for routing; M2 and M3 should be on grid.
- Extend inputs/outputs/power rails to boundary (OUTLINE), and put the labels near the boundary.
- Delay measurements refer to CLK-Q delay, with CLK being WE_MASTER.
- Assume that address is always ready before data.

Submission

Please refer to the submission guidelines mentioned in the CAD1 assignment, as they apply to all CAD assignments. Do not lose points over trivial mistakes like incorrect file names, etc!

For CAD3, you need to submit the following:

- Speedpath:
  - Schematic of the RF speedpath
  - Saved Analog Environment states for speedpath read/write rise & fall delay simulations, named "hspiceD_RFspeed_[read/write]_[rise/fall]". You should submit 4 RF speed states in total (read rise, read fall, write rise, write fall).

  - SPICE waveforms of the speedpath for delay measurement with estimated parasitic capacitance, for all 4 states
  - SPICE waveforms of the speedpath for delay measurement with PEX capacitance OR of the entire RF, for all 4 states
  - SPICE waveforms of the speedpath for setup time/ hold time measurement with PEX capacitance OR of the entire RF
  - *This time is it quite important to submit BOTH waveforms from before and after PEX to demonstrate your estimated loading!
- RF:
  - Schematic and layout of RF
  - DRC/LVS reports
  - NCVerilog waveforms of the entire RF with setup time/hold time/C-Q delay added (write data to all locations in RF & read them out)
- README:
  - List all the full paths of your designs
  - Estimated parasitic capacitance (wire cap) you added to the speed path and the PEX capacitance after layout. How do they compare?
  - List the delays for your speedpath (all 4 states, with estimated capacitance and PEX capacitance) and start/end points of the delay you measured
  - Setup time/hold time for speedpath (with PEX capacitance)
  - Estimate all input pin capacitance (WE, CLK, D, WE_MASTER, READ_A, READ_B) and explain how you estimated those numbers. Compare those to the actual values obtained from PEX.
  - Measure the clock delay after PEX (from CLK to WE0_CLK)
  - Discuss the considerations that went into the choice of your RF design (floor planning, slice width, hierarchy partition, etc).
  - Any other comments you would like us to know.

CAD3 FAQs

Q. Is it ok if the output of the RF is inverted?
A. Yes. But if you'd like, you can also change the design to have non-inverted output. Either is fine, you will just need to design the rest of your datapath to accommodate whichever output polarity you choose.

Q. How to view the post-PEX capacitance value?
A. See the RVE tutorial posted on Canvas.

Deadline

You must complete CAD3 by **January 27, 2022** by **11pm**.
Do not modify any files in your CAD directory after that time.